

## **ELEKTROBOJ**

**Radionica: Mikrokontroleri 101 – Osnove AVR  
ATMEGA8 mikrokontrolera i programiranja u samo 5h**

**KRATKE UPUTE**





## POPIS KOMPONENTI

Provjerite da na stolu imate razvojnu pločicu i pripadajući USB kabel.

## OPIS SKLOPA

Komponenta	Količina
ATMEGA8	1
Podnožje	1
Zen 3.6V	2
R 68 $\Omega$	2
R 1M $\Omega$	1
R 1.5k $\Omega$	1
C 10 $\mu$ F, 6+V	1
C 100n	1
C 22p	2
Q 16MHz	1
Taster	4
LED red, 3mm	1
LED green, 3mm	1
LED yellow, 3mm	1
R 470 $\Omega$	3
LCD 16 $\times$ 2	1
Trimer 10k	1
Pločica 100 $\times$ 160	0.25
UTP kruti	0.25
USB B konektor Ž	1
USB A $\rightarrow$ B kabel	1

Tablica 1: Komponente sklopa

Sklop se napaja preko USB-a i ne treba nikakav drugi izvor napajanja. USB se ujedno koristi i za programiranje mikrokontrolera.

Srce cijelog sklopa je mikrokontroler ATmega8 koji je zadužen za upravljanje svom periferijom (tipkama, svjetlećim diodama i alfanumeričkim pokaznikom), a takt mu je određen kristalom Q1. Priključivanjem na USB ili pritiskom na tipku S1 (reset) mikrokontroler započinje s izvršavanjem programa iz ugrađene flash memorije. Osim flash memorije (8kiB) u mikrokontroleru je i radna memorija (1kiB) te sklopovska podrška za raznu periferiju.

Prije izvršavanja korisničkog programa (na početku flash memorije – 6kiB) pokreće se bootloader (na kraju flash memorije – 2kiB). Bootloader je zadužen za zapisivanje korisničkog programa u flash memoriju preko USB-a. Ukoliko pri svom pokretanju ustanovi da tipka S2 (ISP) nije pritisnuta jednostavno skače na početak korisničkog programa. No ukoliko je prilikom pokretanja tipka S2 pritisnuta započinje programiranje i preko USB-a se prijavi kao AVRasp uređaj. Nakon programiranja mikrokontroler je potrebno resetirati bez pritiskute tipke kako bi skočio u novo zapisani korisnički program.



Mikrokontroler može direktno upravljati izvodima (mijenjati im smjer: ulaz/izlaz i stanje: 1/0) ili indirektno preko neke sklopovske podrške (npr. PWM, serijsko sučelje, ...). Na ovoj radionici koristiti će se samo direktno upravljanje izvodima. Izvode za svjetleće diode je potrebno postaviti kao izlazne, a za tipke kao ulazne s *pull-up* otpornikom.

Alfanumerički pokaznik (display) je cjelina za sebe. Ima ugrađenu podatkovnu memoriju s tekstom kojeg prikazuje te znakovnu memoriju (*bitmap font*) i sam se brine za osvježavanje slike. Mikrokontroler njime upravlja preko 4-bitne podatkovne sabirnice (D4...D7; engl. *data*) i upravljačkih signala RS (engl. *register select*) i E (engl. *enable*). Komunikacijski protokol s pokaznikom izlazi izvan opsega ove radionice. Dovoljno će biti znati koristiti osnovne funkcije iz biblioteke LiquidCrystal koja se sama brine za komunikaciju s pokaznikom.

Shema sklopa je prikazana u dodatku na slici 3.

## PROGRAMIRANJE

Mikrokontroler se programira u razvojnom okruženju Arduino koje u pozadini koristi C++ prevodioc (kompatibilan s C-om). Program započinje izvođenje s funkcijom `setup()` u kojoj se obično nalazi inicijalizacija sklopovlja i ostalih dijelova programa, a nastavlja s beskonačnim pozivanjem funkcije `loop()`.

Posebnu pažnju je potrebno obratiti na način brojanja izvoda. Programski brojevi izvoda **nisu isti** kao i fizički! Slika 2 u dodatku prikazuje vezu između fizičkih i programskih izvoda.

Primjer paljenja zelene svjetleće diode:

```
void setup() {  
  pinMode(11, OUTPUT);  
  digitalWrite(11, HIGH);  
}  
  
void loop() {}
```

Izvode spojene na tipke je potrebno postaviti u način rada `INPUT_PULLUP` kako bi se omogućio interni *pull-up* otpornik u mikrokontroleru na pripadajućem izvodu. Zbog titranja (engl. *bouncing*) je moguće zaključiti da je tipka više puta pritisnuta, no zadatak je takvog tipa da se višestruki pritisci tipke ignoriraju.

Primjer paljenja svjetleće diode dok je pritisnuta tipka:

```
void setup() {  
  pinMode(11, OUTPUT);  
  pinMode(7, INPUT_PULLUP);  
}  
  
void loop() {  
  if (digitalRead(7))  
    digitalWrite(11, HIGH);  
  else  
    digitalWrite(11, LOW);  
}
```

Znakovnom pokazniku se jednostavno pristupa preko biblioteke LiquidCrystal. Prvo je potrebno instancirati objekt razreda LiquidCrystal s argumentima koji određuju lokacije pojedinih upravljačkih izvoda. Zatim je u funkciji `setup()` potrebno pomoću metode `begin(w, h)` započeti



komunikaciju s pokaznikom. Time je završena inicijalizacija pokaznika i moguće je započeti s ispisivanjem teksta. Primjerice:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 0, 1, 3, 5);

void setup() {
  lcd.begin(16, 2);
  lcd.print("Hello!");
  lcd.setCursor(3, 1);
  lcd.print(42);
}

void loop() {}
```

U dodatku se nalazi kratka dokumentacija osnovnih funkcija te metoda razreda LiquidCrystal.

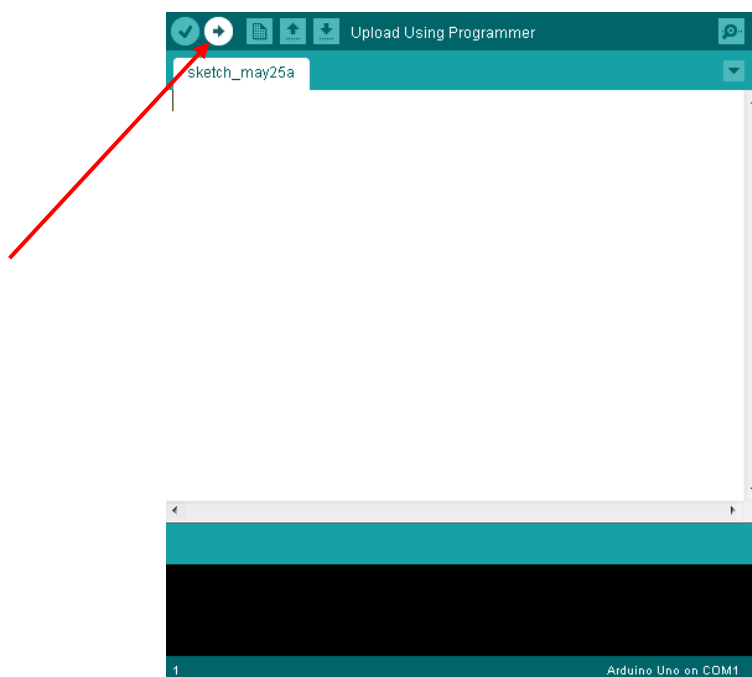
## PODEŠAVANJE ARDUINO PROGRAMA

Kada pokrenete *Arduino* aplikaciju, potrebno je podesiti sljedeće postavke, kako bi mogli *uploadat* Vaš program na mikrokontroler (**Napomena: Ovo se podesi samo jednom!**):

1. *Tools --> Board --> ArduinoNG or older w/ ATmega8*
2. *Tools --> Programmer --> USBasp*

## UPISIVANJE PROGRAMA

1. Priključite razvojnu pločicu na Vaše računalo
2. Pritisnite tipke *RESET* i *PLJUC* zajedno, zatim otpustite tipku *RESET* i odmah nakon nje otpustite i ovu drugu tipku. Nakon ove akcije trebale bi vam na pločici biti upaljene crvena i žuta ledica, što znači da je mikrokontroler spreman za učitavanje Vašeg programa.
3. Držite stisnutu tipku *Shift* te kliknite na strelicu u programu, koja pokazuje prema desno. Kada dodete mišem preko tipke, prije samog klika, kraj nje bi Vam trebalo pisati "*Upload using programmer*". Kada je *upload* uspješno završen, na dnu prozora trebalo bi pisati *Done* i *Verified*. Ukoliko je tokom prevođenja programa došli do grešaka, to će Vam se također ispisati na dnu programa.



**Slika 1: Sučelje Arduino programa**

4. Kada je program uspješno *uploadan*, pritisnite tipku *RESET*, kako bi se Vaš program pokrenuo.

## **ZADACI**

1. Blinky

Treptati LED-icom ili LED-icama po izboru.

2. Gore/dolje, pljuc pljuc!

Implementirajte igricu u kojoj se čovječuljak "i" bori protiv bijesnih majmuna "@".

Čovječuljak se nalazi skroz desno na pokazniku i može ići samo gore-dolje (pomoću S4/S3) ili pljucati zvjezdicu "\*" (pomoću S2) prema majmunu. Samo jednu zvjezdicu može pljucnuti nakon čega mora sačekati ili da pogodi majmuna ili da izađe van pokaznika (samo jedna zvjezdica može biti na pokazniku).

Majmun dolazi s lijeva na desno, kreće s nasumično (random) odabrane visine i nju ne mijenja. Kad izađe iz ekrana započinje idući nivo na kojem dolazi drugi majmun, ali malo brži. Ukoliko se majmun sudari s čovječuljkom, igrač gubi. Igrač ne može pobijediti, ali može se natjecati s drugim igračima tko će do daljeg nivoa doći.

Kada igra završi, potrebno je prikazati nivo do kojeg je igrač došao.



## DODATAK

Kratka dokumentacija funkcija:

- Osnovne funkcije
  - `pinMode(pin, state)`
    - Postavlja način rada izvoda u izlazni, ulazni ili ulazni s *pull-up* otpornikom
    - `pin`: broj izvoda
    - `state`:
      - OUTPUT
      - INPUT
      - INPUT\_PULLUP
  - `digitalWrite(pin, state)`
    - Postavlja logičko stanje izvoda u 1 odnosno 0
    - `pin`: broj izvoda
    - `state`:
      - HIGH/true/1
      - LOW/false/0
  - `digitalRead(pin)`
    - Vraća digitalno stanje izvoda
    - `pin`: broj izvoda
  - `delay(ms)`
    - Čeka zadani broj milisekundi
    - `ms`: broj milisekundi
  - `random(max)`
    - Vraća nasumičan broj između 0 i `max-1`
  - `random(min, max)`
    - Vraća nasumičan broj između `min` i `max-1`
- LiquidCrystal metode
- LiquidCrystal metode
  - `konstruktor(rs, enable, d4, d5, d6, d7)`
    - Inicijalizira sabirnicu za komunikaciju s LCD-om
    - `rs`: broj izvoda RS
    - `enable`: broj izvoda E
    - `d4...d7`: brojevi izvoda viša 4 bita
  - `begin(width, height)`
    - Određuje broj redaka i stupaca LCD-a te inicijalizira LCD
    - `width`: broj stupaca
    - `height`: broj redaka
  - `print(data)`
    - Ispisuje podatak `data` na LCD
    - `data` može biti bilo koji od navedenih tipova:
      - char
      - string (char\*)
      - int
      - long
  - `setCursor(col, row)`
    - Pozicionira kursor na zadane koordinate
    - `col`: broj stupca (počinje od 0)
    - `row`: broj retka (počinje od 0)
  - `clear()`
    - Briše sadržaj LCD-a



[www.arduino.cc](http://www.arduino.cc)

